IN THE CLAIMS:

1.   (Currently Amended) A method in a data processing system for managing data attributes, the method comprising the steps of:

registering attributes with a PKCS9 gateway class, wherein the attributes include user-defined attributes and PKCS-standard (Public Key Cryptography Standards) defined attributes; and

associatively storing an identifier for each of [[said]] the user-defined attributes and each of the PKCS-standard defined attributes.

2.   (Original) The method of claim 1 further comprising:

calling a first object-oriented method in the PKCS9 gateway class, wherein the first object-oriented method receives a parameter comprising an object identifier for an attribute.

3.   ✓ (Original) The method of claim 2 further comprising:

searching an attribute mapping data structure using the object identifier in the received parameter;

in response to a determination of a matching object identifier in the attribute mapping data structure, retrieving a class identifier associatively stored with the matching object identifier in the attribute mapping data structure; and

calling a second object-oriented method in a class identified by the retrieved class identifier.

4.   (Original) A method in a data processing system for managing data attributes, the method comprising the steps of:

invoking a first object-oriented method to process an attribute object, wherein the first object-oriented method is defined in an abstract class for attribute objects with a subclass for undefined attributes and a subclass for defined attributes, wherein the subclass for defined attributes is further comprised of a subclass for each PKCS-defined

(Public Key Cryptography Standards) attribute and a subclass for each user-defined attribute;

invoking a second object-oriented method to process an attribute object, wherein the second object-oriented method is defined in a PKCS9 gateway class; and

in response to invoking the first object-oriented method or the second object-oriented method, processing the result returned by the first object-oriented method or the second object-oriented method.

5.   (Original) The method of claim 4, where a PKCS compatible attribute is a unique object identifier and value, as defined by the Abstract Syntax Notation (ASN.1) for the X.500 Attribute type.

6.   (Original) The method of claim 4, wherein each defined attribute is implemented as a separate class.

7.   (Original) The method of claim 4, wherein each defined attribute is registered with the PKCS9 gateway class.

8.   (Original) The method of claim 7, wherein PKCS-defined attributes are registered statically with the PKCS9 gateway class.

9.   (Original) The method of claim 7, wherein the user-defined attributes are registered with the PKCS9 gateway class by reading a configuration file when the PKCS9 gateway class is initially loaded.

10.   (Original) The method of claim 4, wherein the second object-oriented method determines a type of the attribute object by performing an instanceof comparison to registered attributes.

11.   (Original) The method of claim 4, wherein the attribute object is constructed using a constructor method in a class associated with a PKCS-compatible attribute.

12. (Original) The method of claim 4 wherein, in response to determining an object identifier and a value associated with the object identifier and determining the object identifier is registered with the PKCS9 gateway class, the PKCS9 gateway class returns an instance of a registered attribute.

13. (Original) The method of claim 4 wherein, in response to determining a DER-encoded byte stream with an object identifier, the second object-oriented method in the PKCS9 gateway class returns an instance of a PKCS-compatible attribute.

14. (Original) The method of claim 13 wherein, in response to determining the object identifier from the DER-encoded byte stream is not registered with the PKCS9 gateway class, an instance of an undefined attribute is returned with the value being a DER-encoded byte stream.

15. (Original) The method of claim 13 wherein, in response to determining the object identifier from the DER-encoded byte stream is registered with the PKCS9 gateway class, an instance of an attribute with the object identifier is returned.

16. (Original) The method of claim 4, wherein a registered attribute object is encoded to a DER-encoded byte stream by using the first object-oriented method for encoding the attribute object.

17. (Original) The method of claim 4, wherein a registered attribute object represented as a DER-encoded byte stream is decoded to an attribute object by using the second object-oriented method for decoding the attribute object.

18. (Original) The method of claim 4, wherein a second object-oriented method in the PKCS9 gateway class extracts attribute values into forms, wherein the forms are strings, numbers, and/or other non-abstract data types

19.   (Currently Amended) A data processing system for managing data attributes, the data processing system comprising:

registering means for registering attributes with a PKCS9 gateway class, wherein the attributes include user-defined attributes and PKCS-standard (Public Key Cryptography Standards) defined attributes;

storing means for associatively storing an identifier for each of [[said]] the user-defined attributes and each of the PKCS-standard defined attributes.

20.   (Original) The data processing system of claim 19 further comprising:

calling means for calling a first object-oriented method in the PKCS9 gateway class, wherein the first object-oriented method receives a parameter comprising an object identifier for an attribute.

21.   (Original) The data processing system of claim 20 further comprising:

searching means for searching an attribute mapping data structure using the object identifier in the received parameter;

retrieving means for retrieving, in response to a determination of a matching object identifier in the attribute mapping data structure, a class identifier associatively stored with the matching object identifier in the attribute mapping data structure; and

calling means for calling a second object-oriented method in a class identified by the retrieved class identifier.
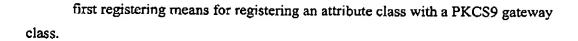
22.   (Original) A data processing system for managing Public Key Cryptography Standards (PKCS) compatible attributes, the data processing system comprising:

first constructing means for constructing a new instance of an attribute object;

first differentiating means for differentiating between attribute objects of different types;

converting means for converting an instance of an attribute object to and/or from DER-encoding;

first extracting means for extracting values associated with an attribute object;

extending means for extending a set of attributes with user-defined types; and
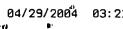
first registering means for registering an attribute class with a PKCS9 gateway class.

23.   (Original) The data processing system of claim 22, where a PKCS compatible attribute is a unique object identifier and value, as defined by the Abstract Syntax Notation (ASN.1) for the X.500 Attribute type.

24.   (Original) The data processing system of claim 22 further comprising:
an abstract attribute object class with an undefined attribute object subclass and a defined attribute object subclass, wherein the defined attribute object subclass is further comprised of a subclass for each PKCS-defined attribute object and a subclass for each user-defined attribute object.

25.   (Original) The data processing system of claim 22 further comprising:
second constructing means for constructing wherein a new instance of an attribute object using a class constructor.

26.   (Original) The data processing system of claim 22 further comprising:
third constructing means for constructing a new instance of a PKCS-compatible attribute object is constructed using the PKCS9 gateway class if an attribute object identifier and a class implementing that attribute are registered.

27.   (Original) The data processing system of claim 22 further comprising:
fourth constructing means for constructing a new instance of a PKCS-compatible attribute object using the PKCS9 gateway class based on a DER-encoded byte stream.

28.   (Original) The data processing system of claim 22 further comprising:
second differentiating means for differentiating a type for an attribute object determined by performing an instanceof comparison to registered attribute classes.

29. (Original) The data processing system of claim 22 further comprising:
encoding means for encoding an attribute object to a DER-encoded object by using an encode method of the attribute object.

30. (Original) The data processing system of claim 22 further comprising:
decoding means for decoding an attribute object represented as a DER-encoded string to an attribute object by using a decode method of the attribute object.

31. (Original) The data processing system of claim 22 further comprising:
second extracting means for extracting the PKCS9 gateway class returns attribute values, wherein the values are represented as strings, numbers, and/or other non-abstract data types.

32. (Original) The data processing system of claim 22 further comprising:
second registering means for registering a PKCS-compatible attribute is registered with the PKCS9 gateway class.

33. (Original) The data processing system of claim 32 further comprising:
third registering means for registering an attribute defined by the Public Key Cryptography Standards is registered with the PKCS9 gateway class.

34. (Original) The data processing system of claim 32 further comprising:
fourth registering means for registering wherein user-defined attributes are registered with the PKCS9 gateway class by reading a configuration file when the PKCS9 gateway class is initially loaded.

35. (Currently Amended) A computer program product in a computer-readable medium for use in a data processing system for managing data attributes, the computer program product comprising:

first instructions for registering attributes with a PKCS9 gateway class, wherein the attributes include user-defined attributes and PKCS-standard (Public Key Cryptography Standards) defined attributes;

second instructions for associatively storing an identifier for each of [[said]] the user-defined attributes and each of the PKCS-standard defined attributes.

36. (Original) The computer program product of claim 35 further comprising:

instructions for calling a first object-oriented method in the PKCS9 gateway class, wherein the object-oriented method receives a parameter comprising an object identifier for an attribute.

37. (Original) The computer program product of claim 36 further comprising:

instructions for searching an attribute mapping data structure using the object identifier in the received parameter;

instructions for retrieving, in response to a determination of a matching object identifier in the attribute mapping data structure, a class identifier associatively stored with the matching object identifier in the attribute mapping data structure; and

instructions for calling a second object-oriented method in a class identified by the retrieved class identifier.

38. (Original) A computer program product on a computer-readable medium for use in a data processing system for managing Public Key Cryptography Standards (PKCS) compatible attributes, the computer program product comprising:

instructions for constructing a new instance of an attribute object;

instructions for differentiating between attribute objects of different types;

instructions for converting an instance of an attribute object to and from DER-encoding;

instructions for extracting values associated with an attribute object; and

instructions for extending a set of attributes with user-defined types; and

instructions for registering an attribute class with a PKCS9 gateway class.

39.   (Original) The computer program product of claim 38, where a PKCS compatible attribute is a unique object identifier and value, as defined by the Abstract Syntax Notation (ASN.1) for the X.500 Attribute type.

40.   (Original) The computer program product of claim 38, wherein an abstract attribute object class with an undefined attribute object subclass and a defined attribute object subclass, wherein the defined attribute object subclass is further comprised of a subclass for each PKCS-defined attribute object and a subclass for each user-defined attribute object.

41.   (Original) The computer program product of claim 38, wherein a new instance of a PKCS-compatible attribute object is constructed using a class constructor.

42.   (Original) The computer program product of claim 38, wherein a new instance of a PKCS-compatible attribute object is constructed using the PKCS9 gateway class if an attribute object identifier and a class implementing that attribute is registered.

43.   (Original) The computer program product of claim 38, wherein a new instance of a PKCS-compatible attribute object is constructed using the PKCS9 gateway class based on a DER-encoded byte stream.

44.   (Original) The computer program product of claim 38, wherein a type for an attribute object is determined by performing an instanceof comparison to registered attribute classes.

45.   (Original) The computer program product of claim 38, wherein an attribute object is encoded to a DER-encoded object by using an encode method of the attribute object.

46.   (Original) The computer program product of claim 38, wherein an attribute object represented as a DER-encoded string is decoded to an attribute object by using a decode method of the attribute object.

47. (Original) The computer program product of claim 38, wherein the PKCS9 gateway class returns attribute values, wherein the values are represented as strings, numbers, and/or other non-abstract data types.

48. (Original) The computer program product of claim 38, wherein a PKCS-compatible attribute is registered with the PKCS9 gateway class.

49. (Original) The computer program product of claim 48, wherein an attribute defined by the Public Key Cryptography Standards is registered with the PKCS9 gateway class.

50. (Original) The computer program product of claim 48, wherein user-defined attributes are registered with the PKCS9 gateway class by reading a configuration file when the PKCS9 gateway class is initially loaded.